Quando criamos uma classe, é possível designar que determinados métodos sejam **estáticos**. Ou seja, estes métodos não são inicializados quando criamos uma nova instância de classe (usando `new`), mas sim a partir da própria classe.

Por exemplo:

```
class User {
 constructor(nome, email, cpf) {
  this.nome = nome
  this.email = email
  this.cpf = cpf
}
exibirInfos() {
  return `${this.nome}, ${this.email}, ${this.cpf}`
}
}
```

No exemplo acima, o método `exibirInfos()` **não** é um método estático, e só é possível executá-lo a partir de uma **instância** da classe `User`:

```
const novoUser = new User('Carol', 'c@c.com', '12312312312')
console.log(novoUser.exibirInfos()) //Carol, c@c.com, 12312312312
```

Se tentarmos executar o método sem associá-lo a nenhuma instância da classe, recebemos um erro:

```
console.log(User.exibirInfos())
  //TypeError: User.exibirInfos is not a function
```

Agora vamos refatorar a classe, declarando `exibirInfos()` como sendo um método estático:

```
class User {
 constructor(nome, email, cpf) {
  this.nome = nome
  this.email = email
  this.cpf = cpf
}
static exibirInfos() {
  return `${this.nome}, ${this.email}, ${this.cpf}`
}
}
```

Ao executarmos, recebemos o seguinte retorno:

```
console.log(User.exibirInfos())
  //undefined, undefined, undefined
```

Não recebemos mais um erro, pois agora o método é `static` e é executado a partir da própria classe, e não de uma instância dela. Porém, como o método depende de informações recebidas do construtor e isso não ocorreu (uma vez que não criamos uma instância

e nem passamos os dados necessários), recebemos `undefined` para cada propriedade.

Vamos fazer mais um teste:

```
class User {
 constructor() {
  this.nome = 'Camila'
  this.email = 'c@c.com'
  this.cpf = '12312312312'
}
 exibirInfos() {
  return `${this.nome}, ${this.email}, ${this.cpf}`
}

static exibeNome(nome) {
  return nome
}
}
```

Mantivemos o método `exibirInfos()` como estava e criamos um novo método, estático, chamado `exibeNome()`. Porém, já vimos que métodos estáticos não podem ser executados a partir de uma instância, então como isso vai funcionar?

```
const novoUser = new User('Carol', 'c@c.com', '12312312312')
const nomeUser = novoUser.nome
console.log(User.exibeNome(nomeUser)) //Camila
```

Criamos uma nova instância de `User` e agora temos acesso à propriedade `nome` desta instância, que estamos chamando de `novoUser`. Agora podemos atribuir a propriedade `novouser.nome` à uma variável (que chamamos de `nomeUser`) e passar o valor dessa variável como parâmetro para a chamada do método estático `User.exibeNome()`.
Na realidade, como `exibeNome()` é um método estático, é possível executá-lo passando qualquer nome como parâmetro:

```
console.log(User.exibeNome('Jaqueline')) //Jaqueline
```

Os métodos estáticos são normalmente utilizados para chamadas de métodos internos de frameworks e bibliotecas, ou em qualquer caso que a classe não dependa de instâncias específicas.

Você pode ver mais sobre esse assunto, com mais exemplos, na documentação do MDN.